View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Information and Computation 192 (2004) 15–40
Information and Computation
www.elsevier.com/locate/ic

On the classification of recursive languages[☆]

John Case,[a,*] Efim Kinber,[b] Arun Sharma,[c] and Frank Stephan[c,1]

[a] *Computer and Information Sciences Department, 101A Smith Hall, University of Delaware, Newark, DE 19716-2586, USA*
[b] *Computer Science, Sacred Heart University, 5151 Park Avenue, Fairfield, CT 06432-1000, USA*
[c] *National ICT Australia Ltd., Sydney Research Laboratory at Kensington, The University of New South Wales, Sydney NSW 2052, Australia*

Received 1 December 1999; revised 27 November 2003
Available online 9 April 2004

Abstract

A one-sided classifier for a given class of languages converges to 1 on every language from the class and outputs 0 infinitely often on languages outside the class. A two-sided classifier, on the other hand, converges to 1 on languages from the class and converges to 0 on languages outside the class. The present paper investigates one-sided and two-sided classification for classes of recursive languages. Theorems are presented that help assess the classifiability of natural classes. The relationships of classification to inductive learning theory and to structural complexity theory in terms of Turing degrees are studied. Furthermore, the special case of classification from only positive data is also investigated.
Crown copyright © 2004 Published by Elsevier Inc. All rights reserved.

1. Introduction

Consider the problem of determining whether a language A over $\mathbb{N}$, the set of natural numbers $\{0, 1, 2, \ldots\}$, satisfies a certain property. Let $\mathcal{A}$ denote the class of all languages over $\mathbb{N}$ that satisfy

[☆] National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

[*] Corresponding author. Fax: 1-302-831-8458.

E-mail addresses: case@cis.udel.edu (J. Case), kinbere@sacredheart.edu (E. Kinber), arun.sharma@nicta.com.au (A. Sharma), frank.stephan@nicta.com.au (F. Stephan).

[1] F. Stephan was previously at the Mathematical Institute, University of Heidelberg, where he was supported by the Deutsche Forschungsgemeinschaft, Heisenberg Grant Ste 967/1–1.

0890-5401/$ - see front matter. Crown copyright © 2004 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2004.03.001

the given property. The question of classification can then be stated thus: if one is given data about A, can one determine if $A \in \mathcal{A}$.

We briefly discuss the various approaches to the study of classification in the literature. One of the earliest attempts was the design of finite automata to decide whether an infinite string (representing the characteristic function of a language) belongs to a given ω-language or not [9,24,36]. But the restrictive computational ability of these finite automata led Büchi [9] and his successors to consider non-deterministic automata. The present paper takes the alternate approach of choosing Turing machines as classifiers. In fact this approach had already been initiated by Büchi and Landweber [10,23].

Smith and Wiehagen [35] introduced a model of classification analogous to the Gold model of learning [8,17,27]. The (recursive) classifier M sees longer and longer prefixes σ of the characteristic function of a language $A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_k$ and guesses on each input σ some number $h \in \{1, 2, \ldots, k\}$ to indicate that $A \in \mathcal{A}_h$. These guesses are supposed to converge, for each set $A \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_k$, to a value h such that $A \in \mathcal{A}_h$. Smith, Wiehagen and Zeugmann [32] extended this study in various ways.

Ben-David [6] and Kelly [22] also interestingly studied classification. They call a class *classifiable* iff there exists a (not-necessarily recursive) functional that indicates in the limit for every A whether or not it belongs to a given class $\mathcal{A}$. They obtained topological conditions for classifiable classes. Gasarch et al. [16] extended this study and obtained relations between the Borel hierarchy on classes—which is induced by the space $\{0,1\}^\infty$ with product topology—and the query hierarchy obtained by allowing a certain number of quantifier-alternations during querying a teacher on the target set A.

Later Stephan [34] investigated the limits of (recursive) classifiers. He considered classification of languages with respect to one single class $\mathcal{A}$ and used the following two natural models of classification: Two-sided classification which is the recursive counterpart to Ben-David's classification in the limit and one-sided classification which is already implicit in the notion of reliable inference (on languages inside the class the learner converges to an index for the language and on languages outside the class the learner makes infinitely many mind changes). These two notions of classes are very natural and coincide with the Δ_2 and Σ_2 classes studied by recursion theorists [18,26,29,30]. Our study derives from these models which we present next. But, first some notation.

We take a *classifier* to be a total recursive function mapping $\{0,1\}^*$ to $\{0,1\}$. We let M, N, and H range over classifiers. Calligraphic letters range over classes, A, B range over sets and U ranges over oracles. We take σ, τ to range over finite strings, interpreted as prefixes of characteristic functions of sets. $\sigma \preceq \tau$ means that $\tau(x){\downarrow} = \sigma(x)$ for all $x \in \mathrm{dom}(\sigma)$; where $\tau(x){\downarrow}$ stands for τ being defined at x and $\tau(x){\uparrow}$ stands for τ being undefined at x. $M(\sigma)$ denotes the guess issued by classifier M on a prefix $\sigma \preceq A$ of the input-set A.

Two-sided classification: For all languages A: $M(\sigma) = \mathcal{A}(A)$ for almost all $\sigma \preceq A$.

Here $\mathcal{A}(A)$ is 1 if $A \in \mathcal{A}$ and 0 otherwise, that is, classes and sets are identified with their characteristic function. "Almost all $\sigma \preceq A$" means "all but finitely many $\sigma \preceq A$." Two-sided classification may be considered to be too strong a requirement. In some applications it is sufficient if the classifier is able to signal the inclusion of a language in a given class, but only provides a weaker signal if the language is not in the class. Stephan [34] introduced the notion of one-sided classification to model this idea.

One-sided classification: For all languages A: if $A \in \mathcal{A}$, then $M(\sigma) = 1$ for almost all $\sigma \preceq A$; if $A \notin \mathcal{A}$, then $M(\sigma) = 0$ for infinitely many $\sigma \preceq A$.

We normally let M and N range over two-sided classifiers and let H range over one-sided classifiers. The notion of one-sided classification is reasonable since the classifier outputs 0 infinitely often thereby guaranteeing that the classifier never locks onto an incorrect conjecture.

In the present paper, we restrict our investigation to classification of recursive languages. Certainly, assuming an algorithmic view of the universe, it is unlikely that nature generates non-recursively enumerable languages. The restriction, then, to recursive languages may be supported by the fact that practical examples in computer science are always recursive. Thus, our classifiers can be relied upon if they are never expected to deliberate upon non-recursive languages. Hence, in the sequel, the statement "for all languages A" in the above two definitions is replaced by "for all recursive languages A." So, we ignore non-recursive sets everywhere and set-theoretic notions like the complement of classes are adapted to the recursive universe: $\overline{\mathcal{A}} = \{\text{recursive } A : A \notin \mathcal{A}\}$.

The present paper may also be seen as closing the gap between Stephan's abstract work [34] and the more concrete approach of Smith et al. [32] and Wiehagen and Smith [35]. Before we begin a formal presentation of the results, we give an informal tour of the various sections in the paper.

In Section 2, we introduce the basic definitions and give preliminary results about two-sided and one-sided classification for classes of recursive languages. We give concrete classes of languages that can be two-sidedly and one-sidedly classified. In particular we observe that one-sided classes are closed under finite monotone Boolean combinations and two-sided classes are closed under all finite Boolean combinations. We also show that every uniformly recursive family of languages is one-sidedly classifiable. Additionally, if the family is discrete, then it is also two-sidedly classifiable. As a consequence of this result, the class of pattern languages is two-sidedly classifiable. As a contrast, however, the class of regular languages is only one-sidedly classifiable.

Although, from [35] we already know that learning and classification are, in general, incomparable, in Section 3, we provide some pleasant links between learning with bounded mind changes and classification. We show that for the concept of exact PEx-learning, a class is identifiable with a (generalized) bound on the number of mind changes iff it is two-sidedly classifiable. For the notion of Ex-learning a weaker version of this correspondence still holds.

In Section 4 we show that classes identifiable in the limit from informants can be *reliably* identified iff they are one-sidedly classifiable. We also investigate conditions under which reliable identification in the limit and two-sided classification are linked.

The characteristic function of a language conveys both positive and negative data about the language. In Section 5, we argue that it may not be realistic to assume the availability of both positive and negative data in practice. The experience from empirical studies of learning is that negative data is not always readily available and even when it is available, it is often tedious to obtain. Motivated by such concerns, we also investigate two-sided and one-sided classification from only positive data. Following the practice in inductive inference literature, we model positive data as texts. As expected, we show that classification from texts is very difficult. As a simple consequence of our result, the class of pattern languages is not even one-sidedly classifiable from texts.

Not deterred by the difficulty of classification from texts, we investigate a weaker version of classification for text presentation, called *partial classification*, that yields some positive results. A class $\mathcal{A}$ is *partially classifiable* just in case there exists a machine that on texts for languages in $\mathcal{A}$

outputs exactly one guess infinitely often and on texts for non-members of $\mathcal{A}$ does not output any single guess infinitely often. The motivation here is that a partial classifier gives a weak signal if the language belongs to the class and refuses to give any signal if the language is not a member of the class being classified. We show that partially classifiable classes can be defined in terms of their index sets: $\mathcal{A}$ is partially classifiable iff its index set $\{e : W_e$ is recursive and $W_e \in \mathcal{A}\}$ is Σ_3. Furthermore, this general criterion turns out to be independent of the data-presentation: the partially classifiable classes are the same for presenting positive data only and presenting positive and negative data at the same time. Most classes considered in practical applications, for example, the class of pattern languages, have a Σ_3 index set and are therefore partially classifiable.

In Section 6, we investigate structurally the computational limits of classifying recursive languages. In particular, we investigate the "computational distance" between one-sided and two-sided classification by determining the kind of non-recursive information that yields a two-sided classifier for a class that was otherwise only one-sidedly classifiable. This gives insight into what it takes for a class of interest to be two-sidedly versus one-sidedly classifiable. We show that access to a high oracle is sufficient to construct a two-sided classifier for a one-sidedly classifiable class. We also establish that in some cases the power of a high oracle is necessary as there are classes for which any two-sided classifier has high Turing degree. We adapt Post's notion of creative set to describe the one-sidedly classifiable classes that are effectively not two-sidedly classifiable. We call a one-sidedly classifiable class $\mathcal{A}$ *creative* just in case there is a uniformly recursive sequence of languages $A_0, A_1, \ldots$ such that for each one-sided classifier H_e, the language A_e is a counterexample to the hypothesis "H_e classifies $\overline{\mathcal{A}}$." The analogy between the two notions of creativity turns out to be quite striking. We give examples of creative classes and show that a creative class is two-sided *only* relative to a high oracle. We discuss some interesting results about one-sidedly classifiable classes of intermediate complexity and compare our results with the more abstract study of classification by Stephan [34] in which a classifier has to behave correctly on non-recursive languages, too.

Finally, in Section 7, we consider classifiers that, instead of guessing 0 or 1, output programs that converge in the limit to 0 or 1. Such programs may be viewed as generators of trial and error guesses and classifiers that output such programs may be viewed to be of somewhat lower quality (compared to the classifiers that directly guess 0 or 1). We consider two kinds of such classifiers: Ex-style requiring that the sequence of programs converge to a single program that has the correct guess of 0 or 1 in the limit and BC-style requiring that the sequence of programs eventually contain only programs that have the correct guess of 0 or 1 in the limit. We show that the notion of Ex-style classification nicely coincides with two-sided classification. We also show that every one-sided classifier has a BC-style classifier. We conclude with insightful, structural characterizations of BC-style classification.

2. Basic definitions and results

Formally, a one-sided classifier is just a $\{0,1\}$-valued function on strings and the languages accepted by this classifier are those where the classifier converges to 1.

Definition 2.1. A classifier H is an algorithm that on every string σ outputs a number 0 or 1. H classifies a class $\mathcal{A}$ *one-sidedly* just in case

- if $A \in \mathcal{A}$, then $H(\sigma) = 1$ for almost all $\sigma \preceq A$;
- if $A \in \overline{\mathcal{A}}$, then $H(\sigma) = 0$ for infinitely many $\sigma \preceq A$.

The classifier H is furthermore *two-sided* iff the statement "for infinitely many" in the second clause can be strengthened to "for almost all". Note that in this definition the variable A ranges over only *recursive* sets. If $\mathcal{A}$ has a one-sided classifier, then $\mathcal{A}$ is called a *one-sided class*, if $\mathcal{A}$ has a two-sided classifier, then $\mathcal{A}$ is called a *two-sided class.* As every two-sided classifier is also one-sided, every two-sided class is also one-sided.

There is an effective list of classifiers H_e such that for each one-sided class there is some H_e classifying it one-sidedly and for each two-sided class there is some H_e classifying it two-sidedly. Let φ_e be an acceptable numbering of all partial recursive functions [26, Section II.5] and assume in the following a fixed numbering $\sigma_0, \sigma_1, \ldots$ of all binary strings. For the following definition of one-sided classifiers H_e, let $\varphi_e(\sigma_x)$ just be interpreted as $\varphi_e(x)$ so that all classifiers can be represented by partial recursive functions.

$$H_e(\sigma) = \begin{cases} \varphi_e(\tau) & \text{for the longest } \tau \preceq \sigma \text{ such that} \\ & \varphi_e(\tau) \text{ outputs 0 or 1 within } |\sigma| \text{ steps;} \\ 0 & \text{if there is no such } \tau; \end{cases}$$

The advantage of the H_e compared to the φ_e is that they are more well-behaved while still having most desired properties with respect to universalness. In particular, the following properties hold.

- The H_e are a uniformly recursive family of total functions $\{0,1\}^* \to \{0,1\}$;
- Every H_e is a one-sided classifier for some class which is called $\mathcal{H}_e$ from now on;
- If φ_e (interpreted as a function on binary strings) is a one-sided classifier for a class $\mathcal{A}$ then $\mathcal{H}_e = \mathcal{A}$.
- If φ_e is two-sided so is H_e. In particular, every two-sided class has a two-sided classifier H_e.

In the sequel, we will consider H_e instead of the underlying φ_e as a list of all potential classifiers and the two-sided classifiers among these H_e play a similar role as the total recursive functions within the list of all partial recursive functions φ_e. The φ_e will stand for normal functions $\mathbb{N} \to \mathbb{N}$ in the sequel since whenever reference is needed to some acceptable system of one-sided classifiers, the system of the H_e will be used.

One-sided classes are closed under finite monotone Boolean combinations and two-sided classes are closed under all finite Boolean combinations. Although this fact is a direct corollary from that fact that one-sided and two-sided classification are the Σ_2 and Δ_2 classes restricted to recursive sets, the proofs are nevertheless included for the sake of completeness.

Fact 2.2. *A class $\mathcal{A}$ is two-sided iff $\mathcal{A}$ and $\overline{\mathcal{A}}$ are one-sided classes. If classes $\mathcal{A}, \mathcal{B}$ are one-sided, so are $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$. If a class $\mathcal{A}$ is one-sided, so is the class $\mathcal{B} = \{B : B$ is a finite variant of some $A \in \mathcal{A}\}$.*

Proof. The direction ($\Rightarrow$) of the first statement is obvious. For the reverse direction ($\Leftarrow$), let H' be a one-sided classifier for $\mathcal{A}$ and let H'' be one for $\overline{\mathcal{A}}$. Let $M(\lambda) = 0$; having already defined $M(\sigma)$ for some $\sigma \in \{0,1\}^*$, we extend the definition inductively to $M(\sigma a)$ for $a = 0, 1$ as follows:

$$M(\sigma a) = \begin{cases} H'(\sigma a) & \text{if } H'(\sigma a) \neq H''(\sigma a); \\ M(\sigma) & \text{otherwise.} \end{cases}$$

We claim that M is a two-sided classifier for $\mathcal{A}$: If a recursive set A is in $\mathcal{A}$, then H' converges on A to 1 while H'' outputs on A infinitely many 0s. So there are infinitely many $\tau \preceq A$ with $H'(\tau) = 1$ and $H''(\tau) = 0$ but only finitely many $\tau \preceq A$ with $H'(\tau) = 0$ and $H''(\tau) = 1$. So M will converge to 1. Similarly M will converge to 0 on any recursive set $A \in \overline{\mathcal{A}}$.

For the second statement, let H' be a one-sided classifier for $\mathcal{A}$ and H'' be a one-sided classifier for $\mathcal{B}$. Now $\mathcal{A} \cap \mathcal{B}$ has the one-sided classifier

$$H(\sigma) = \begin{cases} 1 & \text{if } H'(\sigma) = 1 \text{ and } H''(\sigma) = 1; \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that H outputs on A infinitely many 0s iff either H' or H'' does. So H converges on A to 1 iff both H' and H'' converge on A to 1. The case $\mathcal{A} \cup \mathcal{B}$ is a bit more involved. The following fact is used in defining the machine:

H outputs on A in total at least n 0s if H' and H'' both output on A in total at least n 0s.

This informal idea can be turned into an algorithm as follows: let

$$n_M(\sigma) = |\{\tau \preceq \sigma : M(\tau) = 0\}|$$

for each machine $M \in \{H, H', H''\}$, $H(\lambda) = 1$ and

$$H(\sigma w) = \begin{cases} 0 & \text{if } n_{H'}(\sigma w) > n_H(\sigma) \text{ and } n_{H''}(\sigma w) > n_H(\sigma); \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to see that H outputs infinitely many 0s iff both H' and H'' output infinitely many 0s. Therefore, H converges on a set A to 1 if at least one of the machines H' and H'' converges to 1.

The classifier for $\mathcal{B}$ in the last statement is constructed such that it outputs on B at least n 0s iff the classifier for $\mathcal{A}$ outputs on each set of the form $a_0 a_1 \dots a_n B(n+1) B(n+2) \dots$ at least n 0s. $\square$

Given a recursive function $A(x, y)$, let $A_x = \{y : A(x, y) = 1\}$ and $\mathcal{A}$ be the class $\{A_0, A_1, \dots\}$. Such a class $\mathcal{A}$ is called a *uniformly recursive family*. Angluin [3] initiated the study of learning uniformly recursive families from texts. After the introduction of monotonicity constraints many papers have considered the learnability of these families from texts and informants [20,37,38]. A class $\mathcal{A}$ is *closed* iff for each $A \notin \mathcal{A}$ there is a $\sigma \preceq A$ such that no $B \in \mathcal{A}$ extends σ. The first part of the next fact is also an immediate consequence of the fact, that uniformly recursive families are Σ_2 classes [29]: $A \in \mathcal{A} \Leftrightarrow (\exists x)\,(\forall y)\,[A(y) = A_x(y)]$.

Fact 2.3. *Every uniformly recursive family is one-sided. If it is also closed, then it is two-sided.*

Proof. As in the proof of Fact 2.2 let $n_H(\sigma) = |\{\tau \preceq \sigma : H(\tau) = 0\}|$, $H(\lambda) = 1$ and

$$H(\sigma w) = \begin{cases} 1 & \text{if } \sigma w \preceq A_x \text{ for } x = n_H(\sigma); \\ 0 & \text{otherwise.} \end{cases}$$

The intuitive idea behind H is to check the sets $A_0, A_1, \dots$; whenever A_x turns out to be different from A, H outputs a 0 and moves on to A_{x+1}, otherwise H outputs 1 as long as A_x and A appear to be equal. So H converges on every set A_x to 1 making (at most) x 0s and outputs infinitely many 0s for all $A \notin \mathcal{A}$.

Assume now the same algorithm for a closed class $\mathcal{A}$ and let $A \notin \mathcal{A}$ be recursive. Then there is $\tau \preceq A$ such that no A_x extends τ. In particular $\sigma w \not\preceq A_x$ for all $\sigma \succeq \tau$, all x and almost all $\sigma w \preceq A$. If follows that $H(\sigma w) = 0$ for almost all $\sigma w \preceq A$. So H is already a two-sided classifier for $\mathcal{A}$. $\square$

Example 2.4. The immediately preceding results yield the following examples.

- $\mathcal{C} = \{A : A \text{ is cofinite}\}$ is one-sided, but not two-sided.
  The classifier is $H(\sigma w) = w$.
- $\mathcal{D} = \{1^\infty, 01^\infty, 001^\infty, 0001^\infty, \ldots\}$ is two-sided.
  The classifier M outputs 1 if $\sigma \in 0^*1^+$ and 0 otherwise.
- $\mathcal{E} = \{A : A \text{ has finite and even cardinality}\}$ is one-sided, but not two-sided.
  The classifier $H(\sigma)$ outputs 1 iff the number of 1s in σ is even and 0 iff this number is odd.
- $\mathcal{F}_\phi = \{A : \text{the formula } \phi(A) \text{ is true}\}$ is two-sided.
  Here $\phi(A)$ means that ϕ is a Boolean combination of atomic expressions of the form $c \in A$ for constants c and A being the only free variable representing the input-set A of the same name. For example, $\phi(A) = [5 \in A \vee [3 \notin A \wedge 4 \notin A]]$. Such formulas $\phi(A)$ can be evaluated after having seen a sufficiently long part of the input and from then on the classifier outputs 1 if $\phi(A)$ holds and 0 if $\phi(A)$ does not hold.
- $\mathcal{G} = \{\text{graph}(p) : p \text{ is a polynomial}\}$ is one-sided, but not two-sided.
  $\mathcal{G}$ and $\mathcal{R}$ below are uniformly recursive families and, hence, have the one-sided classifier from Fact 2.3.
- $\mathcal{P} = \{A : A \text{ is a pattern language}\}$ is two-sided.
  This is due to the fact that the class of the pattern languages is both closed and uniformly recursive.
- $\mathcal{R} = \{A : A \text{ is regular}\}$ is one-sided, but not two-sided.

There is also a prominent class which is not one-sided: the class $\{A : \varphi_{\min(A)} \text{ computes } A\}$ of the self describing sets. But this class has a one-sided complement. Furthermore, note that the class $\mathcal{D}$ is uniformly recursive and two-sided but not closed. So the converse direction of the implication in Fact 2.3 does not hold.

3. Two-sided classification with bounded mind changes

An Ex-learner finds for every set A in the class $\mathcal{A}$ to be learned in the limit an index e such that φ_e computes the characteristic function of A. More formally, $\mathcal{A}$ is Ex-learnable iff

$$(\forall A \in \mathcal{A})\ (\exists e)\ (\forall^\infty \sigma \preceq A)\ [M(\sigma) = e \wedge \varphi_e = A],$$

where $\varphi_e = A$ stands for "φ_e computes the characteristic function of A." Each output $M(\sigma)$ for some $\sigma \preceq A$ is called a *guess* or a *hypothesis* of M for A. If $M(\sigma a) \neq M(\sigma)$ (with $\sigma \in \{0,1\}^*$ and $a \in \{0,1\}$) then one says that M makes a mind change. While Gold [17] only required that for each $A \in \mathcal{A}$ the number of mind changes is finite and thus M stabilizes on a hypothesis after reading finitely many data-bits, Bārzdiņš and Freivalds [7] introduced the notion of bounded mind changes where the number of mind changes has to respect a uniform constant bound for all $A \in \mathcal{A}$. Case and Smith [14] applied this notion also to notions like learning with anomalies. Freivalds and Smith [15] generalized the concept of mind change bounds by using constructive ordinals [30]; this concept is equivalent to the second type in the hierarchy below. See [2,19] for examples of natural classes learnable with ordinal mind change bounds.

The basic idea to implement mind change bounds is to equip the learner with a counter and to require that this counter must change its current value to a lower one whenever the learner makes a mind change, that is, $\mathrm{Count}_M(\sigma a) = \mathrm{Count}_M(\sigma)$ if $M(\sigma a) = M(\sigma)$ and $\mathrm{Count}_M(\sigma a) < \mathrm{Count}_M(\sigma)$ if $M(\sigma a) \neq M(\sigma)$. In the more concrete cases, the counter starts with a natural number, say 5, and goes down with every mind change, for example, from 5 to 4, then from 4 to 3 and so on. But whenever the counter reaches 0, it cannot be decreased further and therefore the total number of mind changes is at most 5. The more abstract realization of Freivalds and Smith [15] use counters which range over constructive ordinals. These can be counted down only finitely often since the ordinals are well-ordered and thus enforce that the learner converges to a hypothesis always. In the context of inductive learning Sharma et al. [31] have considered the following four types of bounded mind changes whose definitions can be directly adapted for classification and where Q denotes the permitted range of counter values.

first type: constant bound. The counter can take as value only a natural number.

second type: ordinal bound. The counter is a rational number which is always member of a well-ordered subset Q of the rationals.

third type: linear bound. The set Q of permitted rational numbers has no recursive decreasing infinite sequence but may have a non-recursive one.

fourth type: general bound. The set Q of permitted counter values is equipped with a partial ordering $\sqsubseteq$ such that there are no recursive infinite descending chains.

A fifth type where Q may have a infinite recursive descending chain is just the same as having no restriction at all. In the context of classification, this fifth type is equivalent to one-sided classification. Note that each such definition requires that the set Q of possible counter values is recursively enumerable and the ordering on Q is recursive with domain Q.

Theorem 3.1. *For a class $\mathcal{A}$ the following is equivalent*:
(a) *$\mathcal{A}$ can be classified using a classifier with bounded mind changes of the second type*;
(b) *$\mathcal{A}$ can be classified using a classifier which converges on every set—also on every non-recursive set*;
(c) *$\mathcal{A}$ can be classified using a classifier of the fourth type with the additional requirement that the partially ordered set Q of counter-values does not have any decreasing infinite sequence—also no non-recursive one*.

The proof of this theorem is similar to that in the case of inductive inference [31]; so it is omitted here. The next theorem, however, is quite different to that setting since there are inferable classes which do not have bounded mind changes of the fourth type; an example for such an inferable class is the class of all finite sets.

Theorem 3.2. *Every two-sided class has a mind change bound of the fourth type*.

Proof. Let M be a two-sided classifier for $\mathcal{A}$. Then let Q contain the string λ and all strings σa with $\sigma \in \{0,1\}^*$, $a \in \{0,1\}$ and $M(\sigma a) \neq M(\sigma)$. The ordering $\sqsubset$ is just given by $\tau \sqsubset \sigma$ iff τ is a proper extension of σ. Furthermore M assigns to each mind change on each input σa the string $\sigma a \in Q$ as the counter for the mind change. It remains to show that Q has no descending recursive sequence: Assume that $\sigma_0 \sqsubset \sigma_1 \sqsubset \cdots$ would be such a sequence. Each string σ_k has length at least k and all σ_k coincide on their common domain: so they have a recursive limit A given by $A(x) = \sigma_{x+1}(x)$. M makes at "the end" of each σ_k a mind change, thus M makes infinitely many mind changes on A in contradiction to the fact that M is a two-sided classifier for A. So the theorem follows. □

The next theorem again transfers directly from the case in inductive inference [31].

Theorem 3.3. *The hierarchy given by the four notions of bounded mind changes is proper.*

For the restrictive notion of Popperian explanatory identification (PEx), a strong relation between learnability with bounded mind changes and classifiability is established. Recall that PEx-learning [11,14] means that the learner Ex-identifies the class with the additional requirement that any guess—also on illegal data—is an index of a total recursive function. This has some consequences as the one that it can be checked immediately whether a guess is consistent with the data seen so far, that is, whether $\varphi_e(x)\downarrow = A(x)$ for all those x where the learner has already seen the data $A(x)$. Furthermore M learns a class $\mathcal{A}$ exactly iff M converges only on sets in $\mathcal{A}$ to a correct program; on recursive sets outside $\mathcal{A}$ either M diverges or converges to a program which computes something else.

Theorem 3.4. *Let $\mathcal{A}$ be exactly* PEx*-learnable. Then $\mathcal{A}$ is two-sided iff $\mathcal{A}$ can be exactly* PEx*-learned with bounded mind changes of the fourth type.*

Proof. For the first direction let N be an exact PEx-learner for $\mathcal{A}$ which respects a mind change bound of the fourth type. In other words, N outputs always indices of total functions, N converges on every recursive set A to some hypothesis e and this hypothesis e is an index of the characteristic function of A iff $A \in \mathcal{A}$. Now the classifier M is based on checking whether the $N(\sigma)$th recursive function coincides with the already known part of the characteristic function of the set to be classified.

$$M(\sigma) = \begin{cases} 1 & \text{if } \varphi_{N(\sigma)}(x) = \sigma(x) \text{ for all } x \in \mathrm{dom}(\sigma); \\ 0 & \text{otherwise, that is, } \varphi_{N(\sigma)}(x) \neq \sigma(x) \text{ for some } x \in \mathrm{dom}(\sigma). \end{cases}$$

Since the guesses $N(\sigma)$ are always *total* programs, M is recursive. Furthermore N converges on every recursive set to a fixed program and so M also converges on this recursive set either to 1 if this last program is a program for the set or to 0 otherwise. So the convergence of M follows from that of N. By the exactness, M classifies $\mathcal{A}$ correctly.

For the other direction let M be a two-sided classifier and N be a PEx-learner for $\mathcal{A}$. Furthermore let e be an index of some function in $\mathcal{A}$. Now the new exact PEx-learner respecting the mind change bound of the fourth type works as follows:

$$H(\sigma) = \begin{cases} N(\sigma) & \text{if } M(\sigma) = 1; \\ e & \text{otherwise, that is, } M(\sigma) = 0. \end{cases}$$

If $A \in \mathcal{A}$ then H converges on A to the same program as N and so identifies A. If $A \notin \mathcal{A}$ then M converges on A to 0 and so H on A to e. So H converges on all recursive sets (the others are not considered for mind change bounds of the fourth type). Furthermore H converges to a program for some set in $\mathcal{A}$ while $A \notin \mathcal{A}$, so H is exact. $\square$

This result could be improved to stating that $\mathcal{A}$ is two-sided via a classifier respecting mind change bounds of the kth type iff $\mathcal{A}$ is exactly PEx-learnable via a machine respecting mind change bounds of the kth type. Nevertheless the result depends on the fact that every PEx-learnable class is one-sided. Replacing PEx by Ex, only a weaker version holds.

Theorem 3.5. *Every two-sidedly classifiable and* Ex-*learnable class can be* Ex-*learned with bounded mind changes of the fourth type, but some two-sidedly classifiable and* Ex-*learnable class $\mathcal{A}$ cannot be* Ex-*learned with bounded mind changes of second type.*

Proof. The construction of the PEx-learner H from a two-sided classifier M and an PEx-learner N in the previous theorem can be carbon copied in order to construct an Ex-learner H from a two-sided classifier M and an Ex-learner N. Again H respects mind change bounds of the fourth type.

For the second statement, consider a *simple* set $S = \{a_0, a_1, \ldots\}$. Recall that a simple set is a recursively enumerable set whose complement is infinite but does not contain any infinite recursive set [28]. Let $\mathcal{A} = \{A : |A| \text{ is finite and even and } A \subseteq \overline{S}\}$. A two-sided classifier on input σ checks first whether $\sigma(a_k) = 1$ for some $a_k \in dom(\sigma)$ with $k \leqslant |\sigma|$. If so, then the classifier outputs 0. Otherwise the output is 1 if the number of all x with $\sigma(x)\downarrow = 1$ is even and is 0 if this number is odd. Since no infinite recursive set is disjoint from S, this two-sided classifier for $\mathcal{A}$ is correct.

On the other hand $\mathcal{A}$ cannot be learned with bounded mind changes of second type. Let M be an inductive inference machine which learns $\mathcal{A}$ satisfying a mind change bound of the second type. For each set $A \in \mathcal{A}$ let $q(A)$ be the minimal q_i output during the inference of A. The set $\{q(A) : A \in \mathcal{A}\}$ has a minimum q_j since it is well-ordered. $q_j = q(A)$ for some fixed set A. Now A has finite and even cardinality and there is some $\sigma \preceq A$ such that $M(\sigma)$ is an index for A and M has output q_j while reading this σ. Since $\overline{S}$ is infinite there are $x, y \in \overline{S} - A - \mathrm{dom}(\sigma)$ and M has to infer $A \cup \{x, y\}$. Since also $\sigma \preceq A \cup \{x, y\}$, M has to make a mind change after σ and also output a rational $q_i < q_j$. So $q(A \cup \{x, y\}) < q_j$ in contradiction to the choice of q_j and such a machine M does not exist. $\square$

The following result shows a connection of exact learning with bounded mind changes and one-sided classification. The converse of Theorem 3.6 does not hold since the class of all self-describing sets is a counter-example as mentioned in Example 2.4.

Theorem 3.6. *If $\mathcal{A}$ can be exactly* Ex-*learned with bounded mind changes of the fourth type, then $\overline{\mathcal{A}}$ is one-sided. Furthermore some class, namely the class of all self-describing sets, is* Ex-*learnable with bounded mind changes of the first type but is not one-sided.*

Proof. Assume that M is a learner for $\mathcal{A}$ which respects mind change bounds of the fourth type, that is, which converges on every recursive set A to some program; this program is a program for A iff $A \in \mathcal{A}$. Now the classifier H for $\overline{\mathcal{A}}$ is specified as follows:

H outputs at least n 0s for an input-set A iff there is $m \geqslant n$ such that the index $e = M(A(0)A(1)\dots A(m))$ computes the first $n+1$ values of A: $\varphi_e(0)\downarrow = A(0), \varphi_e(1)\downarrow = A(1), \dots, \varphi_e(n)\downarrow = A(n)$.

It does not matter that this condition is not recursive but recursively enumerable only, since H is not required to output the nth 0 immediately but only after some suitable time. If H outputs finitely many 0s, then M does not converge to an index e which computes A; so H classifies A correctly to be inside $\overline{\mathcal{A}}$. If H outputs infinitely many 0s, then M converges to a program e which coincides on arbitrary long prefixes with A, thus computes A. So H classifies again correctly A to be outside $\overline{\mathcal{A}}$.

The class $\{A : \varphi_{\min(A)}$ computes $A\}$ of the self describing sets from Example 2.4 can be learned with bounded mind changes, namely without any mind change: The learner waits for the first 1 to appear in the characteristic function and then outputs the position of this 1 as an index for A. In the following, the proof that $\mathcal{A}$ is not one-sided, is included since this fact was mentioned but not proven in Example 2.4.

The index set of the self describing sets is Π_2: e is an index of a self describing set if φ_e is total and φ_e outputs 0 on input $x < e$ and 1 on input e. It can be shown that this index set is also Π_2-complete.

But if $\mathcal{A}$ would be one-sided via some recursive machine M then also the complement of the index set of all self describing functions would be Π_2 yielding a contradiction: Given e, e is not the index of a self-describing function iff M outputs infinitely many 0s on every set beginning with 0^e1, that is, iff for each n there is an m such that M outputs on all strings in $0^e1 \cdot \{0,1\}^m$ at least n 0s. □

4. Reliable learning and one-sided classification

A learner M is said to *reliably* Ex-learn a class $\mathcal{A}$ [25] of languages if M converges on all $A \in \mathcal{A}$ and furthermore, whenever M converges on a recursive language A to an index e (whether $A \in \mathcal{A}$ or not), then the function φ_e is the characteristic function of the set A. There are further variants of reliable learning, but we picked the above definition where the learner has to be reliable only on the *recursive sets* while nothing is said about its behaviour on the non-recursive sets. Furthermore, M exactly REx-learns $\mathcal{A}$ iff M diverges on all recursive sets in $\overline{\mathcal{A}}$. The reader may observe that an exactly reliably Ex-learned class is also in some sense classified since convergence indicates membership in the class and divergence indicates membership in its complement. Hence, it might be expected that there are interesting links between reliable learning and classification.

Theorem 4.1. *Let $\mathcal{A}$ be* Ex*-learnable. Then $\mathcal{A}$ is exactly* REx*-learnable iff $\mathcal{A}$ is one-sided.*

Proof. ($\Rightarrow$): Let $\mathcal{A}$ be exactly REx-learnable. The classifier outputs 0 if the learner changes its mind and outputs 1 if there is no mind change. Whenever the learner converges to an index, then the classifier outputs only finitely many 0s and thus accepts the language. Whenever the learner does not converge to an index, that is, the language does not belong to $\mathcal{A}$, then the classifier rejects the language by outputting infinitely many 0s. So the classifier accepts just the languages in $\mathcal{A}$ and is correct.

($\Leftarrow$): If $\mathcal{A}$ is Ex-learnable and one-sidedly classifiable, then a mind change can be introduced into the learning algorithm by padding at every place where the classifier outputs 0, that is, if the learner outputs for σ and σw the same guess e, but the classifier outputs a 0 for σw, then the learner's output at σw is replaced by an equivalent but different index for the characteristic function computed by e. This does not effect convergence on $A \in \mathcal{A}$ since there these new mind changes are inserted only finitely often. But if $A \notin \mathcal{A}$, then the classifier outputs infinitely many 0s which induce infinitely many mind changes on the modified learner; so this modified learner diverges. Thus the modified learner is reliable, that is, it converges on a recursive A if and only if it learns A. Furthermore, the learner is exact since it learns exactly the languages in $\mathcal{A}$. □

The reader may have observed that in the above proof of Theorem 4.1, at no point the guesses are evaluated. Therefore this result can be translated to similar notions as long as the following two conditions are satisfied: padding is available and infinitely many mind changes (as in the notion of behaviourally correct learning) are not permitted.

Case et al. [12] introduced the notion of learning limiting recursive programs or "trial-and-error-guesses" in the context of learning functions. We adapt this notion to identification in the limit of limiting programs for characteristic functions of recursive languages. According to this criterion of learning, a learner has to converge on the characteristic function of every language $A \in \mathcal{A}$ to a total program e in two variables which computes A in the limit:

$$(\forall x)\,(\exists y)\,(\forall z > y)\,[A(x) = \varphi_e(x,z)].$$

We denote by LimEx the class of recursive language classes that can be learned in the above sense. Reliable LimEx identification is then a variant of the above criterion in which the learner is required to converge to a limiting program for every $A \in \mathcal{A}$ and has to make infinitely many mind changes for all recursive $B \notin \mathcal{A}$. Since LimEx satisfies the two conditions above, Theorem 4.1 also holds for learning limiting recursive programs.

Theorem 4.2. *Let $\mathcal{A}$ be in* LimEx. *Then $\mathcal{A}$ is exactly reliably* LimEx *learnable iff $\mathcal{A}$ is one-sided.*

The definitions of reliable learning and bounded mind changes conflict since the first one requires infinitely many mind changes on sets outside $\mathcal{A}$ while the second one requires also in this case finitely many mind changes. This artificial conflict can be removed via permitting a special symbol "?" to indicate non-convergence which also does not count as additional mind change: an exact REx-learner respecting ordinal mind change bounds of the fourth type would then on recursive sets $A \in \mathcal{A}$ converge to an index e for A and on recursive sets $A \notin \mathcal{A}$ converge to the special symbol? Such a learner can be directly translated into a two-sided classifier which outputs 1 if the learner outputs a hypothesis and which outputs 0 if the learner outputs? So one gets the following corollary.

Corollary 4.3. *If $\mathcal{A}$ can be exactly* REx-*learned by a machine with bounded mind changes of the fourth type, then $\mathcal{A}$ is two-sidedly classifiable. Furthermore, every two-sided class $\mathcal{A} \in$* Ex *is exactly* REx-*learnable with bounded mind changes of the fourth type.*

5. Classification from only positive data

Within all previous sections, the source of information was the characteristic function of the set A to be classified, that is, the input for the classifier at stage n is the prefix $A(0)A(1)\cdots A(n)$ of the characteristic function of A. Besides this mode of data-presentation, called *informant*, learning theorists also consider often the case, where the classifier or learner sees a *text*, that is a list of the elements of A as input. In order to compare classification from text with the standard classification in the other sections, this standard method is referred to as "classification from informant" within the present section.

Gold [17] introduced this notion of identification from text which is also known as "identification from positive data" [3]. Formally a *text for a language A* is an infinite sequence of numbers and the symbol "#" such that each element of A appears at least once and no non-element of A ever appears in the sequence. Analogously to Gold's notion of inference, we can define classification from texts: a one-sided classifier for $\mathcal{A}$, upon being fed a text for some language A, converges to 1 iff $A \in \mathcal{A}$. As in the case of standard classification, one has that a two-sided classifier satisfies the additional constraint that it converges to 0 on every text of every recursive set $A \notin \mathcal{A}$.

Example 5.1. Every class $\mathcal{F}_\phi$ of all languages satisfying the formula $\phi(A)$ as defined in Example 2.4 is two-sidedly classifiable from text.

Proof. The classifier is relatively easy and for each input σ evaluates $\phi(\text{range}(\sigma))$. Since ϕ accesses the set A only at a finite number of places, all sufficiently long $\sigma \preceq T$ for a given text T satisfy $x \in \text{range}(\sigma) \Leftrightarrow x \in A$ for the x where ϕ evaluates $A(x)$. For example, if $\phi(A) = (3 \in A \wedge 4 \notin A)$, then all sufficiently long $\sigma \preceq T$ satisfy $3 \in \text{range}(\sigma) \Leftrightarrow 3 \in A$ and $4 \notin \text{range}(\sigma) \Leftrightarrow 4 \notin A$. So the result of evaluating ϕ on range(σ) for these σ is the same as for evaluating ϕ on A. □

Theorem 5.2. *If $\mathcal{A}$ and $\mathcal{B}$ are both two-sidedly classifiable from text and a finite set belongs to $\mathcal{A}$ iff it belongs to $\mathcal{B}$, then $\mathcal{A} = \mathcal{B}$.*

Proof. Assume that $\mathcal{A}$ and $\mathcal{B}$ are both two-sidedly classifiable from text, that each finite set belongs to $\mathcal{A}$ iff it belongs to $\mathcal{B}$ and that A is an infinite and recursive set. Furthermore, let M_1 classify $\mathcal{A}$ and M_2 classify $\mathcal{B}$ from text and let $a_0, a_1, \ldots$ be a recursive enumeration of A. Now define inductively over k a text $T = a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots$ such that $M_1(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots a_k\#^{n_k}) = M_2(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots a_k\#^{n_k})$ for all k; the numbers n_k must all exist since M_1 and M_2 classify each finite set $\{a_0, a_1, a_2, \ldots, a_k\}$ in the same way and thus converge on each text $a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots a_k\#^\infty$ to the same value. So both, M_1 and M_2, take on T infinitely often the same value and both converge on T; therefore both converge to the same limit-value and A is in $\mathcal{A}$ iff A is in $\mathcal{B}$. □

One might ask whether the following chain-condition on two-sidedly classifiable $\mathcal{A}$ must hold.

> Whenever an ascending chain $A_0 \subset A_1 \subset \cdots$ belongs to $\mathcal{A}$ so does some infinite set.

The following counterexample gives a negative answer to this question.

Example 5.3. Consider the class $\mathcal{A} = \{A : A \cap S = \emptyset\}$ where S is a simple set, that is, where S is a recursively enumerable set with an infinite complement that does not contain any infinite recursive subset. This class $\mathcal{A}$ contains an infinite ascending chain of finite sets but no infinite recursive set.

Proof. The simple set S has a recursive enumeration $a_0, a_1, \ldots$ and the two-sided classifier M just checks whether the text seen so far intersects an approximation of S:

$$M(\sigma) = \begin{cases} 0 & \text{if } a_k \in \text{range}(\sigma) \text{ for some } k \leqslant |\sigma|; \\ 1 & \text{otherwise.} \end{cases}$$

Now let $\overline{S} = \{b_0, b_1, \ldots\}$ (where the sequence $b_0, b_1, \ldots$ is of course not recursive). Then $\{b_0\}, \{b_0, b_1\}, \{b_0, b_1, b_2\}, \ldots$ forms this ascending chain of sets in $\mathcal{A}$. But $\mathcal{A}$ has no infinite member since every infinite and recursive set intersects S. □

Furthermore, Theorem 5.2 does not hold for one-sided classification. An example is $\mathcal{A}$ as the class of all finite sets and $\mathcal{B}$ as the class of all sets. Obviously $\mathcal{B}$ can be classified one-sidedly from text by always outputting 1. For $\mathcal{A}$ the algorithm is a bit more difficult: $H(\lambda) = 1$ and $H(\sigma w)$ is 1 if $w \in \text{range}(\sigma)$ and 0 if $w \notin \text{range}(\sigma)$. Thus if the text is for an infinite set, then infinitely often a new element is added and so H outputs infinitely often a 0. If the text is for a finite set, then only finitely often w is a new element and so the classifier converges to 1.

Theorem 5.4. *There is no non-empty class $\mathcal{A}$ that is one-sidedly classifiable from text and contains only infinite languages. In particular the class $\mathcal{P}$ of all pattern-languages is not classifiable from text.*

Proof. Assume that $\mathcal{A}$ contains an infinite set $A = \{a_0, a_1, \ldots\}$, but $\mathcal{A}$ does not contain finite sets. Furthermore, let H be a classifier which is correct on texts of all finite sets. Then there is a text $T = a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots$ such that $H(a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots a_k\#^{n_k}) = 0$ for all k since H must output on each text $a_0\#^{n_0}a_1\#^{n_1}a_2\#^{n_2}\cdots a_k\#^{\infty}$ for each finite set $\{a_0, a_1, \ldots, a_k\}$ infinitely many 0s. So A has a text T such that H outputs on T infinitely many 0s. Thus H is not a one-sided classifier for $\mathcal{A}$ and $\mathcal{A}$ is not one-sided.

The adaptation to $\mathcal{P}$ uses the fact that there is an infinite pattern language A and that every pattern language which contains two different elements already is infinite. Thus the construction to show that $\mathcal{P}$ is not one-sided via H starts with $a_0a_1\#^{n_1}$ and then proceeds in the same way. □

Indeed the construction can be strengthened to prove the existence of some kind of locking-set: If $\mathcal{A}$ can be one-sidedly classified from text and if $A \in \mathcal{A}$ is infinite, then there is a finite set $F \subseteq A$ such that every recursive set B with $F \subseteq B \subseteq A$ belongs to $\mathcal{A}$. Similarly if $\mathcal{A}$ is two-sidedly classifiable from text, also each infinite set $A \notin \mathcal{A}$ has a locking set $F \subseteq A$ such that no recursive set B between F and A ($F \subseteq B \subseteq A$) belongs to $\mathcal{A}$. Using this fact it is possible to show that one almost natural property does not hold for classification from text: Infinite one-sided classes sometimes do not have infinite two-sided subclasses.

Theorem 5.5. *The infinite class $\mathcal{A} = \{\{0, 1, \ldots, a\} : a \in \mathbb{N}\}$ is one-sidedly classifiable from text but every subclass $\mathcal{B} \subseteq \mathcal{A}$ which is two-sidedly classifiable from text is finite.*

Proof. First, it is necessary to show that $\mathcal{A}$ is one-sidedly classifiable from text. This is witnessed by the one-sided classifier M given by $M(\lambda) = 0$ and

$$M(\sigma w) = \begin{cases} 0 & \text{if } w \notin \text{range}(\sigma) \text{ or range}(\sigma w) \text{ is not of the form } \{0, 1, \ldots, a\}; \\ 1 & \text{otherwise.} \end{cases}$$

Let $\mathcal{B}$ be a subclass of $\mathcal{A}$ which is two-sidedly classifiable from text. Then $\mathbb{N}$ is not in $\mathcal{B}$ since $\mathbb{N}$ is not in $\mathcal{A}$ and $\mathbb{N}$ has a locking set F. F has a maximum b and so no set $\{0, 1, \ldots, a\}$ with $a \geqslant b$ is in $\mathcal{B}$; thus $\mathcal{B}$ is finite. □

The preceding theorems showed the limitations of classifying from text. So it is suitable to look for a weaker convergence criterion in order to make it possible to classify more realistic classes from text.

Definition 5.6. A machine H classifies a class $\mathcal{A}$ *partially from text* iff H on any text T for any set A outputs an infinite sequence of numbers such that $A \in \mathcal{A}$ iff exactly one number appears in the output infinitely often and $A \notin \mathcal{A}$ iff no number appears in the output infinitely often.

It is easy to see that every class which can be one-sidedly classified from texts can also be partially classified from texts. But there are classes which can be partially classified but cannot be one-sidedly classified from text. Indeed the partially classifiable classes $\mathcal{A}$ can be characterized in terms of their index sets $\{e : W_e \in \mathcal{A}\}$. Here W_e is the eth recursively enumerable set with respect to a fixed acceptable numbering of all recursively enumerable sets, for example, with respect to the numbering of the domains of the partial recursive functions: $W_e = \{x : \varphi_e(x)\downarrow\}$. Since the intersection of two Σ_3 sets is a Σ_3 set and since the index set $\{e : W_e \text{ is recursive}\}$ is a Σ_3 set, one can ignore those indices e where W_e is not recursive. Theorem 5.7 holds indeed even also for all classes of recursively enumerable sets; it is to a certain extent just the counterpart of the fact, that the class of all recursively enumerable sets can be learned from text under the criterion of partial identification [27].

Theorem 5.7. *A class $\mathcal{A}$ is partially classifiable from text iff its index set $\{e : W_e \in \mathcal{A}\}$ is Σ_3.*

Proof. ($\Rightarrow$): Assume that M classifies $\mathcal{A}$ partially from text. The sets W_e have all a uniform enumeration $a_{e,0}, a_{e,1}, \ldots$ which might contain the pause symbol # in order to deal with the empty set and indices where it is unknown whether the corresponding set is empty or not. Now the predicate

$$(\exists x)\,(\forall y)\,(\exists z > y)\,[M(a_{e,0}a_{e,1}\ldots a_{e,z}) = x],$$

witnesses that the index set of $\mathcal{A}$ is Σ_3.

($\Leftarrow$): For the converse direction one uses the fact that one can make the first quantifier to have either one or zero solutions. More precisely, there is a recursive predicate P such that

$$W_e \in \mathcal{A} \Leftrightarrow (\exists \text{ exactly one } x)\,(\forall y)\,(\exists z)\,[P(e, x, y, z)];$$
$$W_e \notin \mathcal{A} \Leftrightarrow (\forall x)\,(\exists y)\,(\forall z)\,[\neg P(e, x, y, z)].$$

The predicate P can be obtained by using an m-reduction f from the Σ_3 set $\{e : W_e \in \mathcal{A}\}$ to $\{e : W_e \text{ is cofinite}\}$ [33, Corollary IV.3.5] and then letting

$$P(e, x, y, z) \;\Leftrightarrow\; x, x+1, \ldots, x+y \in W_{f(e),\,z} \wedge (x = 0 \vee x-1 \notin W_{f(e),y}).$$

Furthermore, let E be a recursively enumerable set which contains for every recursively enumerable set A exactly one index e with $W_e = A$. Such a set exists since one can obtain it as the range of the translation of a Friedberg numbering into the given acceptable numbering. Now one defines that a classifier H outputs a number coding the pair $\langle e,x\rangle$ at least n times iff there is a prefix σ of the text such that

- $|\sigma| \geqslant n$ and e is enumerated into E within $|\sigma|$ steps;
- $\text{range}(\sigma) \cap \{0,1,\ldots,n\} = W_{e,|\sigma|} \cap \{0,1,\ldots,n\}$;
- $(\forall y \leqslant n)\ (\exists z \leqslant |\sigma|)\ [P(e,x,y,z)]$.

If a number representing the pair $\langle e,x\rangle$ appears infinitely often in the output of H, then the first and second conditions imply that the given text is a text for W_e and $e \in E$ while the third condition implies that x witnesses that $W_e \in \mathcal{A}$. So only sets $A \in \mathcal{A}$ are considered by H to be in $\mathcal{A}$ and it remains to show that H indeed outputs on texts of them one number infinitely often.

Let $A \in \mathcal{A}$ and T be a text for A. Furthermore, let e be the index of $\mathcal{A}$ in E and x be the corresponding unique number such that $(\forall y)\ (\exists z)\ [P(e,x,y,z)]$. For given n, let $\sigma \preceq T$ be so long that e is enumerated into E within $|\sigma|$ stages, all elements of A up to n have appeared in σ and have been enumerated into $W_{e,|\sigma|}$ and every $y \leqslant n$ has a witness $z \leqslant |\sigma|$ such that $P(e,x,y,z)$ holds. Then it follows that $\langle e,x\rangle$ is output at least n times and as a consequence, $\langle e,x\rangle$ is output infinitely often. By the preceding paragraph, any further pair $\langle e',x'\rangle$ is output infinitely often only if $e' \in E$, $W_{e'} = A$ and $(\forall y)\ (\exists z)\ [P(e',x',y,z)]$. As these three conditions imply $e' = e$ and $x' = x$, the uniqueness of the infinitely often output number is guaranteed and the proof is completed. $\square$

Every uniformly recursively enumerable class has a Σ_3 index set, that is, every class for which there is a recursive function f with $\mathcal{A} = \{W_{f(0)}, W_{f(1)}, \ldots\}$. In particular the classes $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{G}$, $\mathcal{P}$, and $\mathcal{R}$ and all classes $\mathcal{F}_\phi$ from Example 2.4 have a Σ_3 index set. Thus they can be partially classified. Assume now that M is a one-sided classifier for $\mathcal{A}$ working on informants. Then

$$\begin{aligned} W_e \in \mathcal{A} &\Leftrightarrow (\exists x)\ (\forall y \geqslant x)\ [M(W_e(0)W_e(1)\cdots W_e(y)) = 1] \\ &\Leftrightarrow (\exists x)\ (\forall y \geqslant x)\ (\forall t)\ (\exists s > t)\ [M(W_{e,s}(0)W_{e,s}(1)\cdots W_{e,s}(y)) = 1], \end{aligned}$$

which is a Σ_3 condition. So $\mathcal{A}$ is partially classifiable from text. The converse does not hold since the class of all infinite languages is not one-sidedly classifiable from informant but has a Σ_3 index set.

Corollary 5.8. *If $\mathcal{A}$ is one-sidedly classifiable from informant, then $\mathcal{A}$ is partially classifiable from text. The converse does not hold.*

So partial classification is very powerful. It is even so powerful, that the method of data-presentation does not matter.

Theorem 5.9. *A class $\mathcal{A}$ can be partially classified from text iff $\mathcal{A}$ can be partially classified from informant.*

Proof. Only the direction "informant $\Rightarrow$ text" has to be shown, since the other one is obvious. Within this proof, the fact that only the behaviour on the recursive sets W_e matters, is crucial—otherwise one could consider a class consisting only of sets of the form $\{2x : x \in A\} \cup \{2y+1 : y \in K\}$ where K is the halting problem; this class could be partially classified from informant iff the index set $\{e : (\exists A)\ [W_e = \{2x : x \in A\} \cup \{2y+1 : y \in K\} \wedge W_e \in \mathcal{A}]\}$ is Σ_4, which is much more powerful than Σ_3. Now let $\mathcal{A}$ be a class with Σ_3 index set. The following predicate assigns to every e an index of the complement e' and can be satisfied only for recursive sets W_e:

$$W_{e'} = \overline{W_e} \Leftrightarrow (\forall x, s)\,(\exists t > s)\,[W_{e',t}(x) + W_{e,t}(x) = 1].$$

Now a set W_e is in $\mathcal{A}$ iff it has a complement $W_{e'}$ and there is a u such that the partial classifier outputs on the characteristic function of W_e—which is verified to be correct using $W_{e'}$— infinitely often this u. Formally,

$W_e \in \mathcal{A} \Leftrightarrow$ there are e' and u such that

- $(\forall x, s)\,(\exists t > s)\,[W_{e',t}(x) + W_{e,t}(x) = 1]$
- $(\forall y)\,(\exists x \geqslant y)\,(\exists t)\,[0, 1, \ldots, x \in W_{e,t} \cup W_{e',t} \wedge M(W_{e,t}(0)W_{e,t}(1) \cdots W_{e,t}(x)) = u]$.

This is an Σ_3 predicate and thus $\mathcal{A}$ is also partially classifiable from text by Theorem 5.7. □

While every class, which is one-sidedly classifiable from text, either contains or is disjoint to an infinite class which is two-sidedly classifiable from text, this does not longer hold for partial classification versus one-sided classification.

Theorem 5.10. *There is a class $\mathcal{A}$ partially classifiable from text such that any infinite class $\mathcal{B}$ which is one-sidedly classifiable from text is neither a subclass of $\mathcal{A}$ nor of $\overline{\mathcal{A}}$.*

Proof. Bi-immune sets are sets E such that neither E nor $\overline{E}$ have an infinite recursive subset. Jockusch [21] showed that such sets exist in every hyperimmune Turing degree, in particular there is a bi-immune Σ_2 set. Relativizing this one obtains that there is a Σ_3 set E such that neither E nor $\overline{E}$ have an infinite Δ_2 subset. Fix such an E. Now the class $\mathcal{A}$ is defined by

$$A \in \mathcal{A} \;\Leftrightarrow \max(A) \text{ exists and is in } E.$$

Note that $\max(A)$ exists iff A is finite and not empty. Thus $\mathcal{A}$ contains all finite non-empty sets A with $\max(A) \in E$.

As the set E is a Σ_3 set, there is a recursive predicate P such that $e \in E$ iff $(\exists x)\,(\forall y)\,(\exists z)\,[P(e, x, y, z)]$. Furthermore, if W_e has a maximum x' then W_e satisfies the condition $(\exists x'')\,(\forall y')\,[\{x'\} \subseteq W_{e,x''+y'} \subseteq \{0, 1, \ldots, x'\}]$. Thus one can combine these formulas and obtain the following characterization:

$$W_e \in \mathcal{A} \Leftrightarrow (\exists x, x', x'')\,(\forall y, y')\,(\exists z)\,[P(e, x, y, z) \wedge \{x'\} \subseteq W_{e,x''+y'} \subseteq \{0, 1, \ldots, x'\}].$$

It follows that $\mathcal{A}$ is partially classifiable from text.

So it remains to show that no infinite one-sided class $\mathcal{B}$ is contained either in $\mathcal{A}$ or in $\overline{\mathcal{A}}$. So let $\mathcal{B}$ be any given class and let H be a one-sided classifier for $\mathcal{B}$. If the subclass $\{B \in \mathcal{B} : B \text{ is finite}\}$ is a finite class, then this subclass is one-sidedly classifiable and coincides with $\mathcal{B}$ by Theorem 5.2. Thus the case that $\mathcal{B}$ has only finitely many finite sets is uninteresting and one from now on considers the case where $\mathcal{B}$ contains infinitely many finite sets. As only finitely many sets of natural numbers have the same maximum, the set $C = \{x : (\exists D \in \mathcal{B})\,[x = \max(D)]\}$ of the maxima of finite sets in $\mathcal{B}$ is infinite. The following formula witnesses that C is a Σ_2 set:

$$x \in C \Leftrightarrow (\exists \eta \in \{0, 1, \ldots, x\}^*)\,(\forall k)\,[H(\eta x^k) = 1].$$

As C is infinite, C has also an infinite Δ_2 subset A. This set A is neither a subset of E nor of $\overline{E}$. So $\mathcal{B}$ has finite sets with maximum inside E and finite sets with the maximum outside E. Thus $\mathcal{B}$ is neither a subclass of $\mathcal{A}$ nor a subclass of $\overline{\mathcal{A}}$. $\square$

A similar result does not hold for one-sided versus two-sided learning from text. Indeed for every one-sidedly text-classifiable class $\mathcal{A}$ there is a subclass $\mathcal{B}$ of either $\mathcal{A}$ or $\overline{\mathcal{A}}$ which is infinite and two-sidedly classifiable from text. This is due to the fact that the set $\{x : \{x\} \in \mathcal{A}\}$ is enumerable relative to K and has either a K-recursive infinite subset or is disjoint to a K-recursive infinite set. This set—call it B in both cases—defines the class $\mathcal{B} = \{\{x\} : x \in B\}$ which is infinite, two-sidedly classifiable from text and is either a subclass of $\mathcal{A}$ or of $\overline{\mathcal{A}}$.

These remarks complete the study of classification from text and for the remaining part of the paper, classification is considered to be classification from informant without explicit notice.

6. Structural properties of classification

Soare [33] contains an extensive study on the relation between recursively enumerable and recursive sets. As Stephan [34] has already noted, the situation of one-sided versus two-sided classification is similar to that of recursively enumerable versus recursive sets. This relationship not only holds in the setting of classifying all sets but also in the setting of the present paper of classifying recursive sets.

This section shows that if only recursive sets are to be classified, then the analogy with recursively enumerable versus recursive sets is even more striking. Turing degrees, an important tool for studying recursively enumerable sets, also turn out to be useful in analyzing the complexity of one-sided classification. The next result shows that—similarly to Stephan's general setting [34]—every one-sided class is two-sided relative to a sufficiently complex oracle.

An oracle U is *Turing reducible* to V (written: $U \leqslant_T V$) iff U can be computed by a machine which has access to a database containing V by the membership-queries "Is $x \in V$?" For an oracle U the relativized halting problem U' to U is defined as $U' = \{e : \varphi_e^U(e)\downarrow\}$ where φ_e^U is the eth partial U-recursive function. U is *high* iff $K' \leqslant_T U'$. Note that this definition differs slightly from Soare's definition [33, Definition IV.4.2] since he considers only oracles $U \leqslant_T K$ and so defined "$K' \equiv_T U'$" instead of "$K' \leqslant_T U'$". An alternative characterization is that there is a function u recursive relative to U which dominates every total recursive function f in the sense that $(\forall^\infty x)\,[u(x) > f(x)]$. Adleman and Blum [1] showed that high oracles play a significant role in inductive inference: The class of all recursive sets is Ex-identifiable relative to U iff U is high. Theorems 6.1 and 6.5 show that the high oracles play a similar special role in classification.

Theorem 6.1. *For each high oracle U, every one-sided class $\mathcal{A}$ has a two-sided classifier which is recursive relative to U.*

Proof. Let H be a one-sided classifier for a class $\mathcal{A}$ of recursive sets. Furthermore let u be a function recursive relative to U which dominates every recursive function. Now the two-sided classifier is defined as follows where $n_H(\sigma)$ denotes as in Fact 2.2 the number of prefixes $\tau \preceq \sigma$ with $H(\tau) = 0$. The idea is now to repeat each 0 of H a large but finite number of times such that M still converges to 1 if H does but M converges to 0 if H only diverges.

If $u(n_H(\sigma)) > |\sigma|$, then let $M(\sigma) = 0$ else let $M(\sigma) = 1$.

If $A \in \mathcal{A}$, then there is only a finite number n of prefixes $\tau \preceq A$ with $H(\tau) = 0$. Almost all prefixes σ of A have length at least $u(n)$. So $|\sigma| \geqslant u(n) \geqslant u(n_H(\sigma))$ and $M(\sigma) = 1$ for these prefixes σ. If $A \notin \mathcal{A}$ and A is recursive, then also the function $f_A(n) = \min\{m : n_H(A(0)A(1)\ldots A(m)) \geqslant n\}$ is recursive and thus u dominates f_A. There is an n with $u(m) > f(m)$ for all $m \geqslant n$. In particular whenever a prefix $\sigma \preceq A$ has at least the length $u(n)$, then $u(n_H(\sigma)) > f_A(n_H(\sigma)) \geqslant |\sigma|$ and $M(\sigma) = 0$. So M converges on every recursive set outside $\mathcal{A}$ to 0 and M is two-sided. □

Nevertheless there are hard problems, that is, there are one-sided classes $\mathcal{A}$ which require that *every* two-sided classifier for $\mathcal{A}$ has high Turing degree. The Turing degree of a machine M is the Turing degree of the set $\{x : M(\sigma) = 1$ for the xth binary string $\sigma\}$ where the definition of this set is based on some canonical enumeration of all binary strings.

Theorem 6.2. *If M two-sidedly classifies the class $\mathcal{C} = \{C : C$ is cofinite$\}$, then M is not recursive and the Turing degree of M is high.*

Proof. Consider the machine H given by $H(\lambda) = 1$ and $H(\sigma a) = a$ for all $\sigma \in \{0,1\}^*$ and $a \in \{0,1\}$. This H is a one-sided classifier for $\mathcal{C}$. Assume now that M is a (not necessarily recursive) two-sided classifier for $\mathcal{C}$. It is shown that the index set $I = \{e : W_e$ is finite$\}$ can be computed relative to M in the limit and thus the Turing degree of M must be high. Let $W_{e,s}$ be a uniformly recursive class of finite sets enumerating the sets W_e and let

$$A_e(s) = \begin{cases} 0 & \text{if } W_{e,s+1} \neq W_{e,s}; \\ 1 & \text{otherwise, that is, } W_{e,s+1} = W_{e,s}. \end{cases}$$

So A_e is cofinite iff W_e is finite, that is, $I(e) = 1$ iff $A_e \in \mathcal{C}$ iff M converges on A_e to 1. Furthermore, the sets A_e are uniformly recursive and so M converges on every set A_e. So $I(e) = \lim_n M(A_e(0)A_e(1)\ldots A_e(n))$ and I is recursive in the limit relative to M. The Turing degree of M is high and in particular, M is not recursive. □

A recursively enumerable set E is called *creative* [33, Definition II.4.3] iff there is an effective procedure which disproves for every e the hypothesis "$W_e = \overline{E}$" by a counterexample $f(e)$, that is, either $f(e) \in \overline{E} - W_e$ or $f(e) \in W_e - \overline{E}$. The name "creative" derives from the fact that such an f creates a new element $f(e) \in \overline{E}$ outside W_e whenever $W_e \subseteq \overline{E}$. This concept is adapted to the context of classifying recursive sets.

Definition 6.3. A one-sidedly classifiable class $\mathcal{A}$ is *creative* iff there is a uniformly recursive array $A_0, A_1, \ldots$ such that for each one-sided classifier H_e the set A_e is a counterexample to the hypothesis "H_e classifies $\overline{\mathcal{A}}$."

The next theorem shows that there is a creative class, namely the class of all cofinite sets. So this class is effectively not two-sided.

Theorem 6.4. *The class $\mathcal{C}$ of all cofinite sets is creative.*

Proof. Let inductively $A_e(0) = 0$ and $A_e(n+1) = H_e(A_e(0)A_e(1)\cdots A_e(n))$. If H_e converges on this set to 1, then it is cofinite and not in $\overline{\mathcal{C}}$. Otherwise H_e does not converge to 1 and outputs infinitely many 0s. Then also A_e is coinfinite and belongs to $\overline{\mathcal{C}}$. So A_e proves that H_e is not an one-sided classifier for the complement of $\mathcal{C}$. $\mathcal{C}$ is creative since its complement is effectively not one-sided. □

All creative sets are 1-equivalent to K and have in particular the same Turing degree as K, that is, belong to the greatest recursively enumerable Turing degree. So it is natural to ask how complex are the creative classes. The next theorem shows that there is indeed an analogous result that only the high oracles allow them to be two-sidedly classified.

Theorem 6.5. *Every creative class is two-sided only relative to high oracles.*

Proof. Let $\mathcal{A}$ be a creative class and let M be a not necessarily recursive two-sided classifier for $\mathcal{A}$. Furthermore, let $A_0, A_1, \cdots$ be a uniformly recursive family of sets such that each A_e witnesses that H_e does not classify $\overline{A}$. Thus $A_e \in \mathcal{A}$ iff H_e converges on A_e to 1.

It is easy to code an infinite array of machines $H_{f(e)}$ such that the machines are independent on the actual input A and that $H_{f(e)}$ outputs on any input A infinitely many 0s iff W_e is infinite. This can be achieved easily by

$$H_{f(e)}(\sigma) = \begin{cases} 0 & \text{if } W_{e,|\sigma|+1} \neq W_{e,|\sigma|}; \\ 1 & \text{otherwise, that is, if } W_{e,|\sigma|+1} = W_{e,|\sigma|}; \end{cases}$$

where one assumes without loss of generality that the approximation $W_{e,s}$ to W_e satisfies $W_{e,s} \subseteq \{0, 1, \ldots, s\}$ and that therefore $|W_{e,s}|$ can be computed from e and s. The classifier $H_{f(e)}(\sigma)$ takes into consideration only the length $|\sigma|$ but not whether $\sigma(x) = 0$ or $\sigma(x) = 1$ for the $x \in \mathrm{dom}(\sigma)$. Therefore $H_{f(e)}$ behaves on all sets A in the same way. Furthermore, if W_e is finite, then these finitely many elements have all shown up at some stage s and $H_{f(e)}(\sigma)$ is 1 for all σ being longer than s. If W_e is infinite, then $|W_{e,s+1}| > |W_{e,s}|$ for infinitely many s and $H_{f(e)}$ outputs on each set A infinitely many 0s. So it holds that

$$\begin{aligned} &W_e \text{ is finite} &&\Rightarrow A_{f(e)} \in \mathcal{A} \Rightarrow M \text{ converges on } A_{f(e)} \text{ to } 1; \\ &W_e \text{ is infinite} &&\Rightarrow A_{f(e)} \notin \mathcal{A} \Rightarrow M \text{ converges on } A_{f(e)} \text{ to } 0. \end{aligned}$$

So using M it can be computed in the limit whether W_e is finite or infinite and thus the Turing degree of M must be high. □

While the preceding results mainly dealt with creative classes, the following one deals with several degrees of non-creativeness. First it is shown that there are one-sided classes of intermediate complexity: they are two-sided relative to some non-high oracle but not relative to the empty oracle. In particular they are also not creative by Theorem 6.5.

Theorem 6.6. *For each U such that K is recursive relative to U and U is recursively enumerable relative to K there is a one-sided class $\mathcal{A}$ such that, for every oracle V, there is a two-sided V-recursive classifier* *for $\mathcal{A}$ iff U is recursive relative to V'. In particular there are intermediate one-sided classes; these are neither two-sided nor creative.*

Proof. Since U is recursively enumerable relative to K, there is a uniformly recursive sequence $U_0, U_1, \ldots$ of sets such that

$$(\forall x)\,[x \in U \Leftrightarrow (\forall^{\infty} y)[x \in U_y]\,].$$

Now let

$$\mathcal{A} = \{A : A = 0^x 1^{\infty} \text{ for some } x \in U\}.$$

Using the sequence $U_0, U_1, \ldots$ it is possible to give the following one-sided classifier for $\mathcal{A}$:

$$M(\sigma) = \begin{cases} 1 & \text{if } \sigma = 0^x 1^y \text{ and } x \in U_y; \\ 0 & \text{otherwise.} \end{cases}$$

If $x \in U$, then x is also in almost all U_y and M outputs 1 on almost all inputs $0^x 1^y$, that is, M classifies $0^x 1^{\infty}$ to be in $\mathcal{A}$. If $x \notin U$, then x is also not in infinitely many U_y and it follows that $M(0^x 1^y) = 0$ for infinitely many y. Thus M classifies $0^x 1^{\infty}$ not to be in $\mathcal{A}$. It is furthermore easy to see that M classifies every input of a form different from $0^x 1^{\infty}$ also not to be in $\mathcal{A}$, so M is indeed a one-sided classifier for $\mathcal{A}$.

Relative to any oracle V, it is equally difficult to classify two-sidedly the set $\{x, x+1, \ldots\}$ (with the characteristic function $0^x 1^{\infty}$) and to compute $U(x)$ in the limit. From this fact it follows that there is some two-sided classifier N recursive relative to V iff U is recursive relative to V in the limit, that is, iff $U \leqslant_T V'$. Since there are oracles U which are enumerable in K and properly between K and K' ($K <_T U <_T K'$), the corresponding classes $\mathcal{A}$ are neither two-sided nor creative. □

Recall that a set is *immune* iff it does not have an infinite recursive subset. For classes, there are two kinds of immunity-properties:

- For a class $\mathcal{A}$ there is no uniformly recursive array $A_0, A_1, \ldots$ of pairwise different sets such that $\{A_0, A_1, \ldots\} \subseteq \mathcal{A}$.
- No infinite two-sided class $\mathcal{B}$ is contained in $\mathcal{A}$.

The following theorems investigate the extent to which one-sided classes and their complements satisfy these requirements. But, the first result shows that a one-sided class and its complement can never be simultaneously immune.

Theorem 6.7. *Let $\mathcal{A}$ be a one-sided class. Then there is a uniformly recursive array $A_0, A_1, \ldots$ of pairwise distinct sets such that the class $\mathcal{B} = \{A_0, A_1, \ldots\}$ is two-sided and either $\mathcal{B} \subseteq \mathcal{A}$ or $\mathcal{B} \subseteq \overline{\mathcal{A}}$.*

Proof. If there is a string σ such that every recursive $A \succeq \sigma$ is in $\mathcal{A}$, then the array given by $A_e = \sigma 0^e 1^{\infty}$ forms a two-sided subclass of $\mathcal{A}$.

Otherwise it is possible to construct such an array outside $\mathcal{A}$. Let H be a one-sided classifier for $\mathcal{A}$. For each e define $A_e = \lim_n \sigma_n$ by $\sigma_0 = 0^e 1$ and σ_{n+1} being the first proper extension of σ_n with

$H(\sigma_{n+1}) = 0$. Since each σ_n is prefix of some $A \notin \mathcal{A}$ there is always such an extension σ_{n+1} and the so defined sets A_e are uniformly recursive. Furthermore, H outputs on every A_e infinitely often 0 and thus no A_e is in $\mathcal{A}$.

It remains in both cases to show that the class $\mathcal{B}$ is two-sided. The algorithm is in both cases the same, the σ in it is the empty string in the second case and the common prefix in the definition of the A_e in the first case. Let

$$M(\tau) = \begin{cases} 1 & \text{if } \sigma 0^e 1 \preceq \tau \preceq A_e \text{ for some } e; \\ 0 & \text{otherwise.} \end{cases}$$

If the set to be classified is some A_e, then M outputs 1 for all inputs beyond the prefix $\sigma 0^e 1$ and thus M converges on A_e to 1. Otherwise the set A to be classified is different from all A_e. Either no $\sigma 0^e 1$ is prefix of A and then M always outputs 0 or some prefix $\sigma 0^e 1 \eta$ is a prefix of A but not of A_e. In this case also $M(\tau) = 0$ for all inputs beyond this prefix. So M always converges to the correct value and is a two-sided classifier for $\mathcal{B}$. □

Theorem 6.8. *There is an infinite two-sided class $\mathcal{A}$ which contains no subclass $\mathcal{B} = \{A_0, A_1, \ldots\}$ consisting of a uniformly recursive array of pairwise distinct sets.*

Proof. Let U be an immune set below K. Then $\mathcal{A} = \{0^x 1^\infty : x \in U\}$ is two-sided but each uniformly recursive array $A_0, A_1, \ldots$ of sets in $\mathcal{A}$ is finite since $\{x : (\exists e)\, [0^x 1 \preceq A_e]\}$ is a recursively enumerable subset of U and therefore finite. □

The next theorem states that there is something analogous to simple sets which are recursively enumerable and coinfinite but intersect every infinite recursive set.

Theorem 6.9. *There is an infinite one-sided class such that its complement has no two-sided infinite subclass.*

Proof. Let U be a set which is enumerable relative to K but whose infinite complement does not have an infinite K-recursive subset, that is, U is a set which is simple relative to K.

Now the class

$$\mathcal{A} = \{A : A \cap U \neq \emptyset\}$$

is infinite but its complement does not have an infinite two-sided subclass. □

It is well-known that every infinite recursively enumerable set has an infinite recursive subset. Stephan [34] showed that this easy observation does not generalize to one-sided classification versus two-sided in his model which requires correct classification of non-recursive sets. Furthermore Theorem 5.5 shows something similar for classification from texts. Since the classification of only recursive sets from informants is more well-behaved than the two previously mentioned settings, the following problem might still have a positive solution.

Problem. Does every infinite one-sided class have an infinite two-sided subclass?

7. Classification by finding trial-and-error programs

Baliga, Case, Jain, Sharma, and Suraj studied in several papers [4,5,12] the concept of learning (or using) limiting or mind-changing programs (equivalently, K-recursive programs) instead of ordinary programs for classes of recursive functions. This concept transfers quite naturally to classification: Instead of guesses 0 and 1, the classifier produces a sequence of programs such that each of these programs converges in the limit to either 0 or 1 which then stands for the guess of the classifier. More formally such a classifier assigns to every input σ a primitive recursive program e such that $L(e) = \lim_n \varphi_e(n)$ exists and is either 0 or 1. As in inductive inference there are two notions of convergence.

- Ex-*style classification*: For every recursive set A, the classifier outputs for almost all $\sigma \preceq A$ the same guess e and $L(e) = \mathcal{A}(A)$.
- BC-*style classification*: For every recursive set A, the classifier outputs for almost all $\sigma \preceq A$ an index e_σ such that $L(e_\sigma) = \mathcal{A}(A)$.

Theorem 7.1. Ex-*style classification and two-sided classification coincide*.

Proof. It is easy to see that outputting a constant 0 or 1 can be transferred into outputting a program which converges in the limit to 0 or 1, respectively. So only the direction to transfer an Ex-style classifier into an two-sided classifier for the same class is interesting. Given an Ex-style classifier M the new two-sided classifier N is defined by $N(\sigma) = \varphi_{M(\sigma)}(|\sigma|)$. Since M always outputs indices of primitive recursive functions, N is total. Assume now that A is recursive. Then M outputs for almost all $\sigma \preceq A$ the same index e. Furthermore $\varphi_e(n) = \mathcal{A}(A)$ for almost all n. It follows that $N(\sigma) = \mathcal{A}(A)$ for almost all $\sigma \preceq A$. $\square$

Theorem 7.2. *Every one-sided class has a* BC-*style classifier*.

Proof. By Theorem 6.1 every one-sided class is classifiable two-sidedly relative to a high oracle, in particular it has a K-recursive classifier M. By the Limit-Lemma [26, Proposition IV.1.17], there is a primitive recursive function N such that $M(\sigma) = \lim_x N(\sigma, x)$. Using the Substitution-Theorem there is a primitive recursive procedure assigning to each σ an index $e(\sigma)$ for the function $f(x) = N(\sigma, x)$. This index $e(\sigma)$ is then the output of the BC-style classifier which classifies the same sets as M. $\square$

It is easy to see that the concept of BC-style classification is closed under complementation. Thus the inclusion of one-sided classification into a BC-style classifier is proper. The proof of Theorem 7.2 showed already that every class which is two-sided relative to the oracle K is already BC-style classifiable. This can be extended to a characterization of BC-style classification by the following theorem.

Theorem 7.3. *For a class $\mathcal{A}$ of recursive sets the following is equivalent*:
(a) *$\mathcal{A}$ is* BC-*style classifiable*.
(b) *$\mathcal{A}$ is two-sided relative to K*.
(c) *Both index sets $\{e : W_e$ is recursive and $W_e \in \mathcal{A}\}$ and $\{e : W_e$ is recursive and $W_e \in \overline{\mathcal{A}}\}$ are Σ_3*.

Proof. It suffices to show (a $\Rightarrow$ c) and (c $\Rightarrow$ b).

Given a BC-style classifier M, the following Σ_3 predicate describes the set of all e on which M converges to c:

$$\mathcal{A}(W_e) = c \Leftrightarrow (\exists x)\,(\forall y \geqslant x)\,(\forall s)\,(\exists t \geqslant s)\,[\varphi_{M(W_{e,t}(0)W_{e,t}(1)\ldots W_{e,t}(y))}(t) = 1].$$

Since M converges on every recursive set $\mathcal{A}$, the intersection of these predicates with a Σ_3 predicate defining that W_e is recursive gives the transition from (a) to (c).

For every Σ_3 set E there is a K-recursive predicate P such that $e \in E \Leftrightarrow (\exists x)\,(\forall y)\,[P(e,x,y)]$. Let P_0 and P_1 the corresponding P's for the Σ_3 formulas for membership and non-membership of recursive sets W_e in $\mathcal{A}$. Now the K-recursive two-sided classifier M for $\mathcal{A}$ works as follows:

> On input σ, mark all (e,x) as disqualified if either $W_e(y) \neq \sigma(y)$ for some $y \in dom(\sigma)$ or $(\exists y, y' \leqslant |\sigma|)\,[\neg P_0(e,x,y) \wedge \neg P_1(e,x,y')]$. Find the first pair (e,x) not yet disqualified and output 0 if $(\forall y \leqslant |\sigma|)\,[P_0(e,x,y)]$ and 1 otherwise.

For any recursive set A, the following holds: If $W_e \neq A$ then all pairs (e,x) are eventually disqualified. If $W_e = A$, then there is an x such that either $P_0(e,x,y)$ or $P_1(e,x,y)$ holds for all y. Then this pair (e,x) is never disqualified and thus, for almost all $\sigma \preceq A$, there is the same first pair (e,x) which is not disqualified. So there is a unique $c \in \{0,1\}$ such that $(\forall y)\,[P_c(e,x,y)]$—this c is just $\mathcal{A}(A)$. If σ is long enough, then also the y with $\neg P_{1-c}(e,x,y)$ is bounded by $|\sigma|$ and $M(\sigma)$ outputs c. So M is a two-sided K-recursive classifier for $\mathcal{A}$. $\square$

So it follows that every BC-style classifiable class is also partially identifiable, but the converse does not hold since there are classes $\mathcal{A}$ where $\{e : W_e$ is recursive and $W_e \in \mathcal{A}\}$ is Σ_3 but $\{e : W_e$ is recursive and $W_e \notin \mathcal{A}\}$ is not.

Acknowledgments

We thank the referees for numerous comments, detailed proofreading, and suggesting the current title. A previous version of the paper has been presented at the Symposium on Theoretical Aspects of Computer Science [13].

References

[1] L.M. Adleman, M. Blum, Inductive inference and unsolvability, The Journal of Symbolic Logic 56 (1991) 891–900.

[2] A. Ambainis, S. Jain, A. Sharma, Ordinal mind change complexity of language identification, Theoretical Computer Science 220 (1999) 323–343.

[3] D. Angluin, Inductive inference of formal languages from positive data, Information and Control 45 (1980) 117–135.

[4] G. Baliga, J. Case, Learning with higher order additional information, in: Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII 1994, 5th International Workshop on Algorithmic Learning Theory, ALT 1994, Reinhardsbrunn Castle, Germany, October 10–15, 1994, Proceedings, Springer Lecture Notes in Artificial Intelligence 872, 1994, pp. 64–75.

[5] G. Baliga, J. Case, S. Jain, M. Suraj, Machine learning of higher order programs, The Journal of Symbolic Logic 59 (1994) 486–500.

[6] S. Ben-David, Can finite samples detect singularities of real-valuedfunctions? Algorithmica 22 (1998) 3–17.
[7] J. Bārzdiņš, R. Freivalds, Prediction and limiting synthesis of recursively enumerable classes of functions, Latvijas Valsts Universitate Zinatnieski Raksti 210 (1974) 101–111.
[8] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Information and Control 28 (1975) 125–155.
[9] J.R. Büchi, On a decision method in restricted second order arithmetic (in Russian), Kiberneticheskij Sbornik 8 (1964) 78–90.
[10] J.R. Büchi, L.H. Landweber, Definability in the monadic second-order theory of successor, The Journal of Symbolic Logic 34 (1969) 166–170.
[11] J. Case, S. Jain, Suzanne Ngo Manguelle, Refinements of inductive inference by Popperian and reliable machines, Kybernetika 30 (1994) 23–52.
[12] J. Case, S. Jain, A. Sharma, On learning limiting programs, International Journal of Foundations of Computer Science 3 (1992) 93–115.
[13] J. Case, E. Kinber, A. Sharma, F. Stephan, On the classification of computable languages, in: STACS 1997, 14th Annual Symposium on Theoretical Aspects of Computer Science, Lübeck, Germany, February 27–March 1, 1997, Proceedings, Springer Lecture Notes in Computer Science, 1200, 1997, pp. 225–236.
[14] J. Case, C.H. Smith, Comparison of identification criteria for machine inductive inference, Theoretical Computer Science 25 (1983) 193–220.
[15] R. Freivalds, C.H. Smith, On the role of procrastination in machine learning, Information and Computation 107 (1993) 237–271.
[16] W. Gasarch, M.G. Pleszkoch, F. Stephan, M. Velauthapillai, Classification using information, Annals of Mathematics and Artificial Intelligence 23 (1998) 147–168.
[17] E.M. Gold, Language identification in the limit, Information and Control 10 (1967) 447–474.
[18] P.G. Hinman, Recursion-Theoretic Hierarchies. Perspectives in Mathematical Logic, Springer, Heidelberg, 1978.
[19] S. Jain, A. Sharma, Elementary formal systems, intrinsic complexity and procrastination, Information and Computation 132 (1997) 65–84.
[20] K.P. Jantke, Monotonic and non-monotonic inductive inference, New Generation Computing 8 (1991) 349–360.
[21] C. Jockusch, The degrees of bi-immune sets, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 15 (1969) 135–140.
[22] K. Kelly, The Logic of Reliable Inquiry. Logic and Computation in Philosophy, Oxford University Press, New York, 1995.
[23] L.H. Landweber, Decision problems for ω-automata, Mathematical Systems Theory 3 (1969) 376–384.
[24] R. McNaughton, Testing and generating infinite sequences by a finite automaton, Information and Control 9 (1966) 434–448.
[25] E. Minicozzi, Some natural properties of strong identification in inductive inference, Theoretical Computer Science 2 (1976) 345–360.
[26] P. Odifreddi, Classical Recursion Theory. Studies in Logic and the Foundations of Mathematics, 125, North-Holland, Amsterdam, 1989.
[27] D.N. Osherson, M. Stob, S. Weinstein, Systems That Learn, Bradford/MIT Press, London, 1986.
[28] E. Post, Recursively enumerable sets of positive integers and their decision problems, Bulletin of the American Mathematical Society 50 (1944) 284–316.
[29] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill Book Company, New York, 1967.
[30] G.E. Sacks, Higher Recursion Theory. Perspectives in Mathematical Logic, Springer, Heidelberg, 1990.
[31] A. Sharma, F. Stephan, Y. Ventsov, Generalized notions of mind change complexity, Information and Computation 189 (2004) 235–262.
[32] C.H. Smith, R. Wiehagen, T. Zeugmann, Classifying predicates and languages, International Journal of Foundations of Computer Science 8 (1997) 15–42.
[33] R.I. Soare, Recursively Enumerable Sets and Degrees. Perspectives in Mathematical Logic, Springer, Heidelberg, 1987.
[34] F. Stephan, On one-sided versus two-sided classification, Archive for Mathematical Logic 40 (2001) 489–513.

[35] R. Wiehagen, C.H. Smith, Generalization versus classification, Journal of Experimental and Theoretical Artificial Intelligence 7 (1995) 163–174.

[36] B.A. Trakhtenbrot, Finite automata and the logic of one place predicates (in Russian), Siberian Mathematical Journal 3 (1962) 103–131.

[37] T. Zeugmann, S. Lange, A guided tour across the boundaries of learning recursive languages, in: Algorithmic Learning for Knowledge-Based Systems, GOSLER Final Report. Springer Lecture Notes in Artificial Intelligence, 961, 1995, pp. 190–258.

[38] T. Zeugmann, S. Lange, S. Kapur, Characterizations of monotonic and dual monotonic language learning, Information and Computation 120 (1995) 155–173.